

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Щёкина Вера Витальевна
Должность: Ректор
Дата подписания: 2019.05.22:14:56:46
Уникальный программный ключ:
a2232a55157e67a551a8799b1160192a55989470420556d0f573a454e37789

	МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
	Федеральное государственное бюджетное образовательное учреждение высшего образования «Благовешенский государственный педагогический университет»
	ОСНОВНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА Рабочая программа дисциплины

УТВЕРЖДАЮ

**И.о. декана физико-математического
факультета ФГБОУ ВО «БГПУ»**

**О.А. Днепровская
«22» мая 2019 г.**

**Рабочая программа дисциплины
ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Направление подготовки

**02.03.03 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И
АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

Профиль

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

**Уровень высшего образования
БАКАЛАВРИАТ**

**Принята
на заседании кафедры информатики
и методики преподавания информатики
(протокол № 9 от «15» мая 2019 г.)**

Благовешенск 2019

СОДЕРЖАНИЕ

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ	5
3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)	6
4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ	7
5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ	10
6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА.....	10
7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ	21
В ПРОЦЕССЕ ОБУЧЕНИЯ	21
8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	21
9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ	21
10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА	22
11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ	24

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Цель дисциплины: изучение современных методов создания надежного, качественного программного обеспечения, удовлетворяющего предъявляемым к нему требованиям; формирование у студентов понимания необходимости применения данных методов в программной инженерии.

1.2 Место дисциплины в структуре ООП: Дисциплина «Технология разработки программного обеспечения» относится к дисциплинам обязательной части блока Б1 (Б1.О.21).

Для освоения дисциплины «Технология разработки программного обеспечения» используются знания, умения и виды деятельности, сформированные в процессе изучения дисциплин «Программирование», «Структуры и алгоритмы компьютерной обработки данных», «Современные технологии программирования», «Технология программирования Java». Дисциплина «Технология разработки программного обеспечения» в профессиональной подготовке выпускника обеспечивает раскрытие общего круга вопросов разработки программного обеспечения.

1.3 Дисциплина направлена на формирование следующих компетенций: УК-3, ОПК-3, ОПК-4, ПК-4:

– **УК-3.** Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде, индикаторами достижения которой является:

- УК-3.2 – **умеет** строить отношения с окружающими людьми, с коллегами.

– **ОПК-3.** Способен применять современные информационные технологии, в том числе и отечественные, при создании программных продуктов и программных комплексов различного назначения; **индикаторами достижения** которой является:

- ОПК-3.1 – **знает** основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов;

- ОПК-3.2 – **умеет** использовать их в профессиональной деятельности;

- ОПК-3.3 – **имеет практические навыки** разработки программного обеспечения.

– **ОПК-4.** Способен участвовать в разработке технической документации программных продуктов и программных комплексов с использованием стандартов, норм и правил; **индикаторами достижения** которой является:

- ОПК-4.1 – **знает** основные стандарты, нормы и правила разработки технической документации программных продуктов и программных комплексов;

- ОПК-4.2 – **умеет** использовать их при подготовке технической документации программных продуктов;

- ОПК-4.3 – **имеет практические навыки** подготовки технической документации.

– **ПК-4.** Способен применять современные информационные технологии при проектировании, реализации, оценке качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях; индикаторами достижения которой является:

- ПК-4.1 – **знает** современные технологии проектирования и производства программного продукта;

- ПК-4.2 – **умеет** использовать подобные технологии при создании программных продуктов;

- ПК-4.3 – **имеет практический опыт** применения подобных технологий.

1.4 Перечень планируемых результатов обучения. В результате изучения дисциплины студент должен:

- знать:

- основные этапы и методы создания программного продукта;
- основные и вспомогательные процессы программной инженерии;
- преимущества инженерного подхода к созданию программного обеспечения;
- основные сложности, возникающие при внедрении инженерного подхода;
- историю создания и развития программной инженерии;
- связь программной инженерии с жизненным циклом программных средств;

- уметь:

- профессионально грамотно формулировать задачу программирования;
- оформлять программную документацию;
- использовать языки и системы программирования для решения профессиональных задач;
- выполнять необходимое тестирование, отладку или верификацию программы;
- применять информационные технологии при проектировании программного обеспечения;

- владеть:

- методами построения моделей и процессов управления проектами программных средств;
- методами проектирования программного обеспечения, инструментами и методами программной инженерии, языками процедурного и объектно-ориентированного программирования;
- навыками описания программного продукта средствами языка UML.

1.5 Общая трудоемкость дисциплины «Технология разработки программного обеспечения» составляет 4 зачетных единицы (далее – ЗЕ) (144 часа):

Программа предусматривает изучение материала на лекциях и лабораторных занятиях. Предусмотрена самостоятельная работа студентов по темам и разделам. Проверка знаний осуществляется фронтально, индивидуально.

1.6 Объем дисциплины и виды учебной деятельности

Объем дисциплины и виды учебной деятельности (очная форма обучения)

Вид учебной работы	Всего часов	Семестр 5
Общая трудоемкость	144	144
Аудиторные занятия	62	62
Лекции	28	28
Лабораторные занятия	34	34
Самостоятельная работа	46	46
Вид итогового контроля	36	экзамен

2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

2.1 Очная форма обучения

Учебно-тематический план

№	Наименование тем (разделов)	Всего часов	Аудиторные занятия		Самостоятельная работа
			Лекции	Лабораторные занятия	
1.	Программная инженерия. Программное обеспечение	4	2	0	2
2.	Процесс производства: методы, технология и инструментальные средства	10	2	4	4
3.	Жизненный цикл программного обеспечения	4	2	0	2
4.	Анализ предметной области и требования к программному обеспечению	10	2	4	4
5.	Проектирование программного обеспечения; архитектура программного обеспечения; виды диаграмм UML	18	4	6	8
6.	Документирование программного обеспечения	10	2	4	4
7.	Проектирование пользовательского интерфейса	6	2	2	2
8.	Качество программного обеспечения	4	2	0	2
9.	Верификация программного обеспечения	14	4	4	6
10.	Управление разработкой программного обеспечения; организация труда в коллективе и инструментальные средства поддержки	28	6	10	12
Экзамен		36			
ИТОГО		144	28	34	46

Интерактивное обучение по дисциплине

№	Наименование тем (разделов)	Вид занятия	Форма интерактивного занятия	Кол-во часов
1.	Процесс производства: методы, технология и инструментальные средства. Гибкие методологии разработки программного обеспечения	Лб	Дискуссия	2
2.	Анализ предметной области и требования к программному обеспечению	Лб	Работа в малых группах	2

3.	Проектирование программного обеспечения; архитектура программного обеспечения; виды диаграмм UML	Лб	Работа в малых группах	2
4.	Документирование программного обеспечения	Лб	Работа в малых группах	2
5.	Проектирование пользовательского интерфейса	Лек	Лекция-беседа	2
6.	Управление разработкой программного обеспечения; организация труда в коллективе и инструментальные средства поддержки	Лб	Работа в малых группах	2
7.	Управление разработкой программного обеспечения; организация труда в коллективе и инструментальные средства поддержки	Лб	Разработка проекта	2
8.	Верификация программного обеспечения	Лб	Работа в малых группах	2
ИТОГО				16

3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)

Тема 1. Программная инженерия. Программное обеспечение

История возникновения программной инженерии. Термин «программное обеспечение». Программирование и производство программ. Эволюция системного программного продукта по Ф. Бруксу. Неотъемлемые свойства программного обеспечения. Характеристики крупномасштабных программных проектов. Компоненты системы разработки программного обеспечения.

Тема 2. Процесс производства: методы, технология и инструментальные средства

Технология разработки программного обеспечения. Методы, средства и процедуры технологии разработки программного обеспечения. Этапы эволюции технологии разработки программ. Объектный подход к разработке программного продукта. Методологии процесса разработки программного обеспечения.

Тема 3. Жизненный цикл программного обеспечения

Понятие жизненного цикла программного обеспечения. Процессы жизненного цикла программного обеспечения. Модели жизненного цикла программного обеспечения.

Тема 4. Анализ предметной области и требования к программному обеспечению

Виды требований к программному обеспечению. Сбор и анализ требований к программному обеспечению. Верификация и формализация требований к программному обеспечению. Диаграмма вариантов использования.

Тема 5. Проектирование программного обеспечения; архитектура программного обеспечения; виды диаграмм UML

Анализ области решений. Архитектура программного обеспечения. Характеристики программного обеспечения, определяемые архитектурой. Разработка и оценка архитектуры на основе сценариев. UML. Виды диаграмм UML.

Тема 6. Документирование программного обеспечения

Организация документирования программного обеспечения. Виды документации. ГОСТ ЕСПД и другие стандарты, устанавливающие правила разработки программной документации. Стадии разработки программы и программной документации.

Тема 7. Проектирование пользовательского интерфейса

Психологические и физиологические факторы, учитываемые при проектировании пользовательского интерфейса. Основные этапы проектирования пользовательского интерфейса. Факторы оценки удобства использования программного обеспечения. Принципы и методы создания удобного программного обеспечения.

Тема 8. Качество программного обеспечения

Понятие и виды качества программного обеспечения. Стандарты, используемые при разработке программного обеспечения. Характеристики качества программного обеспечения и их атрибуты. Метрики оценки атрибутов качества программного обеспечения. Управление качеством программного обеспечения.

Тема 9. Верификация программного обеспечения

Определение, цели и задачи верификации программного обеспечения. Тестирование программного обеспечения, тестовые данные, тестовая ситуация, удачный тест, ошибка, отказ. Типы процессов тестирования. Тестирование программного кода: задачи и цели, методы, тестовое окружение, драйверы и заглушки, тестовые классы. Тестовые примеры: типы, граничные условия, классы эквивалентности. Тест-план. Покрытие программного кода. Сбой, отказ, авария. Методы разработки устойчивого кода.

Тема 10. Управление разработкой программного обеспечения; организация труда в коллективе и инструментальные средства поддержки

Задачи управления проектом. Виды ресурсов, используемых в проекте. Окружение проекта: организация-исполнитель, заинтересованные в проекте лица. Виды деятельности, входящие в управление проектом. Групповая разработка, управление версиями: единый репозиторий проекта. Организация коллектива разработчиков.

4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

4.1 Общие методические рекомендации

Излагаемая дисциплина предлагает к изучению теоретические основы технологии разработки программного обеспечения и в то же время имеет четкую прикладную направленность. При разработке рабочей программы дисциплины предусмотрено, что определенные вопросы изучаются студентами самостоятельно.

Практикум по дисциплине представлен учебно-методическим материалом по подготовке к лабораторным занятиям. Дидактические материалы для контроля (самоконтроля) усвоения учебного материала содержат вопросы допуска к лабораторным работам и примерные вопросы экзамена. Раздел программы «Список литературы и информационных ресурсов» позволяет использовать материалы не только для подготовки к аудиторным занятиям, но и для организации самостоятельной работы, а также для расширения собственных представлений по отдельным разделам изучаемой дисциплины.

Практикум по дисциплине включает:

- тематику и план лабораторных занятий;
- краткие теоретические и учебно-методические материалы по каждой теме, позволяющие студенту ознакомиться с вопросами, обсуждаемыми на лабораторных работах;

- вопросы допуска для подготовки к лабораторным работам;
- список литературы и информационных ресурсов, необходимых для целенаправленной подготовки студентов к каждому занятию.

Материалы практикума направлены на изучение современных методологий разработки программного обеспечения, предусматривают сочетание аудиторных и внеаудиторных форм организации учебного процесса.

Основное предназначение дидактических материалов – помочь студентам организовать самостоятельную подготовку по дисциплине, провести самоконтроль умений и знаний, получить чёткое представление о предстоящих формах контроля.

4.2 Методические рекомендации по подготовке к лекциям

Курс лекций строится на основе четких понятий и формулировок, так, как только при таком подходе студенты приобретают культуру абстрактного мышления, необходимую для высококвалифицированного бакалавра в любой отрасли знаний. Изложение материала должно быть по возможности простым и базироваться на уровне разумной строгости. Изложение теоретического материала дисциплины должно предшествовать лабораторным занятиям.

4.3 Методические рекомендации по подготовке к лабораторным занятиям

Цель лабораторного практикума – самостоятельная разработка студентами объектно-ориентированной системы с использованием основ современной программной инженерии.

Основными задачами практикума являются:

- проведение анализа предметной области и требований к программной системе;
- создание моделей анализа и проектирования объектно-ориентированных систем с использованием языка моделирования UML;
- проектирование иерархии форм пользовательского интерфейса программной системы;
- знакомство со средствами коллективной разработки программного обеспечения и приобретение навыков работы в команде;
- создание руководства пользователя программного продукта;
- знакомство с существующими стратегиями объектно-ориентированного тестирования и приобретение навыков разработки тестов для конкретного программного продукта.

В ходе проведения практикума студент должен:

- выбрать задачу для разработки объектно-ориентированной программы;
- выполнить двенадцать лабораторных работ, задания на которые размещены в Электронной информационно-образовательной среде БГПУ;
- предоставить отчет по каждой лабораторной работе.

При решении практических задач используются интерактивные методы обучения, позволяющие интенсифицировать процесс понимания, усвоения и творческого применения студентами полученных знаний, повысить мотивацию и вовлеченность их в решение обсуждаемых проблем, что дает эмоциональный толчок к последующей поисковой активности обучающихся, побуждает их к конкретным действиям, процесс обучения становится более осмысленным. Одной из интерактивных форм проведения занятия является работа в малых группах, позволяющая студентам приобрести навыки сотрудничества и другие важные межличностные навыки.

Лабораторный практикум предполагает разбиение студентов на команды по 3-4 человека с распределением ролей руководителя проекта-разработчика, системного аналитика-разработчика и тестера-разработчика, и прохождение каждой командой всех фаз процесса разработки программного продукта.

Каждая команда выбирает задачу из предложенного списка либо предлагает свой вариант темы для проекта. В процессе разработки преподаватель выступает в роли заказчика

программного продукта, студенты из другой малой группы привлекаются в качестве иных заинтересованных лиц. Постановка задачи даётся в краткой форме. Задача студентов – извлечь необходимые сведения из заказчика и остальных заинтересованных в проекте.

Результатом работы команды должен быть работающий прототип программной системы, приложение для тестирования кода и необходимые документы, разработанные в ходе выполнения лабораторных работ.

Выполнение лабораторного практикума по дисциплине, задания которого размещены в Электронной информационно-образовательной среде БГПУ, учитывается при выставлении оценки на экзамене:

- оценка 5 (отлично) ставится, если задания лабораторного практикума выполнены на 90 и более процентов;
- оценка 4 (хорошо) ставится, если задания лабораторного практикума выполнены на 75 и более процентов.
- оценка 3 (удовлетворительно) ставится, если задания лабораторного практикума выполнены на 60 и более процентов.

4.4 Методические указания к самостоятельной работе студентов

Для успешного усвоения дисциплины необходима правильная организация самостоятельной работы студентов. Эта работа должна содержать:

- регулярную (еженедельную) проработку теоретического материала по конспектам лекций и учебникам;
- регулярную (еженедельную) подготовку к лабораторным занятиям, в том числе изучение описания лабораторных работ.

4.5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине:

1. Фонд оценочных средств.
2. Вопросы к экзамену.
3. Список литературы и информационных ресурсов.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

№	Наименование раздела (темы)	Формы/виды самостоятельной работы	Количество часов
1.	Программная инженерия. Программное обеспечение	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ.	2
2.	Процесс производства: методы, технология и инструментальные средства	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к дискуссии.	4
3.	Жизненный цикл программного обеспечения	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ.	2
4.	Анализ предметной области и требования к программному обеспечению	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторной работе.	4

5.	Проектирование программного обеспечения; архитектура программного обеспечения; виды диаграмм UML	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторной работе.	8
6.	Документирование программного обеспечения	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторной работе.	4
7.	Проектирование пользовательского интерфейса	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторной работе.	2
8.	Качество программного обеспечения	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ.	2
9.	Верификация программного обеспечения	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторной работе.	6
10.	Управление разработкой программного обеспечения; организация труда в коллективе и инструментальные средства поддержки	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Подготовка к лабораторной работе.	12
Итого			46

5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ

5.1 План проведения лабораторных занятий по дисциплине

Процесс производства: методы, технология и инструментальные средства.

Лабораторная работа № 1. Гибкие методологии разработки программного обеспечения.

Анализ предметной области и требования к программному обеспечению.

Лабораторная работа № 2. Сбор и анализ требований к программному обеспечению.

Лабораторная работа № 3. Диаграмма вариантов использования.

Проектирование программного обеспечения; архитектура программного обеспечения; виды диаграмм UML.

Лабораторная работа № 4. Диаграмма классов.

Лабораторная работа № 5. Диаграммы последовательности и кооперации.

Лабораторная работа № 6. Диаграмма состояний.

Проектирование пользовательского интерфейса.

Лабораторная работа № 7. Иерархия форм приложения.

Управление разработкой программного обеспечения; организация труда в коллективе и инструментальные средства поддержки.

Лабораторная работа № 8. Средства поддержки коллективной разработки программного обеспечения.

Лабораторная работа № 9. Разработка прототипа программного продукта.

Документирование программного обеспечения.

Лабораторная работа № 10. Руководство пользователя.

Верификация программного обеспечения.

Лабораторная работа № 11. Тест-план.

Лабораторная работа № 12. Тестирование программного продукта.

6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА

6.1 Оценочные средства, показатели и критерии оценивания компетенций

Индекс компетенции	Оценочное средство	Показатели оценивания	Критерии оценивания сформированности компетенций
УК-3, ОПК-3, ОПК-4, ПК-4	Проект	Низкий (неудовлетворительно)	<p>Ответ студенту не зачитывается если:</p> <ul style="list-style-type: none"> • Проект выполнен менее, чем на половину; • Студент демонстрирует отсутствие вовлеченности в групповую работу и невыполнение отведенной ему роли; • Студент обнаруживает незнание большей части теоретического материала, необходимого для реализации проекта; • Студент не сдает отчеты о промежуточных этапах работы над проектом.
		Пороговый (удовлетворительно)	<p>Проект выполнен на 60-80% либо на низком уровне сложности.</p> <ul style="list-style-type: none"> • Студент демонстрирует низкую степень включенности в групповую работу и четкости выполнения отведенной роли; • Студент неполно и с ошибками излагает теоретический материал, необходимый для реализации проекта. • Студент сдает отчеты о промежуточных этапах работы над проектом со значительным опозданием.
		Базовый (хорошо)	<p>Проект выполнен в полном объеме и на достаточном уровне сложности, но с небольшими недочетами.</p> <ul style="list-style-type: none"> • Студент демонстрирует недостаточную степень включенности в групповую работу и четкости выполнения отведенной роли; • Студент допускает незначительные ошибки в трактовке теоретического материала, необходимого для реализации проекта.

			<ul style="list-style-type: none"> • Студент сдает отчеты о промежуточных этапах работы над проектом с небольшим опозданием.
		<p>Высокий (отлично)</p>	<p>Проект выполнен в полном объеме и на достаточном уровне сложности.</p> <ul style="list-style-type: none"> • Студент демонстрирует высокую степень включенности в групповую работу и чёткости выполнения отведенной роли; • Студент обнаруживает понимание теоретического материала, необходимого для реализации проекта. • Студент вовремя сдает отчеты о промежуточных этапах работы над проектом.
<p>УК-3, ОПК-3, ОПК-4, ПК-4</p>	<p>Разноуровневые задачи и задания</p>	<p>Низкий (неудовлетворительно)</p>	<p>Ответ студенту не зачитывается если:</p> <ul style="list-style-type: none"> • Задание выполнено менее, чем на половину; • Студент обнаруживает незнание большей части соответствующего материала, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно излагает материал.
		<p>Пороговый (удовлетворительно)</p>	<p>Задание выполнено более, чем на половину. Студент обнаруживает знание и понимание основных положений задания, но:</p> <ul style="list-style-type: none"> • Излагает материал неполно и допускает неточности в определении понятий; • Не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; • Излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
		<p>Базовый (хорошо)</p>	<p>Задание в основном выполнено. Ответы правильные, но:</p> <ul style="list-style-type: none"> • В ответе допущены малозначительные ошибки и недостаточно полно раскрыто содержание вопроса; • Не приведены иллюстрирующие примеры, недостаточно чётко выражено обобщающее мнение студента; • Допущено 1-2 недочета в последовательности и языковом оформлении излагаемого.
		<p>Высокий (отлично)</p>	<p>Задание выполнено в максимальном объеме. Ответы полные и правильные.</p> <ul style="list-style-type: none"> • Студент полно излагает материал, дает правильное определение основных понятий;

			<ul style="list-style-type: none"> • Обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры; • Излагает материал последовательно и правильно с точки зрения норм литературного языка.
УК-3, ОПК-3, ОПК-4, ПК-4	Доклад, сообщение	Низкий (неудовлетворительно)	<p>Доклад студенту не зачитывается если:</p> <ul style="list-style-type: none"> • Студент не усвоил значительной части проблемы; • Допускает существенные ошибки и неточности при рассмотрении ее; • Испытывает трудности в практическом применении знаний; • Не может аргументировать научные положения; • Не формулирует выводов и обобщений; • Не владеет понятийным аппаратом.
		Пороговый (удовлетворительно)	<p>Задание выполнено более чем на половину. Студент обнаруживает знание и понимание основных положений задания, но:</p> <ul style="list-style-type: none"> • Тема раскрыта недостаточно четко и полно, то есть студент освоил проблему, по существу излагает ее, опираясь на знания только основной литературы; • Допускает несущественные ошибки и неточности; • Испытывает затруднения в практическом применении полученных знаний; • Слабо аргументирует научные положения; • Затрудняется в формулировании выводов и обобщений; • Частично владеет системой понятий.
		Базовый (хорошо)	<p>Задание в основном выполнено:</p> <ul style="list-style-type: none"> • Студент твердо усвоил тему, грамотно и по существу излагает ее, опираясь на знания основной литературы; • Не допускает существенных неточностей; • Увязывает усвоенные знания с практической деятельностью; • Аргументирует научные положения; • Делает выводы и обобщения; • Владеет системой основных понятий.
		Высокий (отлично)	<p>Задание выполнено в максимальном объеме.</p>

			<ul style="list-style-type: none"> • Студент глубоко и всесторонне усвоил проблему; • Уверенно, логично, последовательно и грамотно его излагает; • Опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью; • Умело обосновывает и аргументирует выдвигаемые им идеи; • Делает выводы и обобщения; • Свободно владеет понятиями.
УК-3, ОПК-3, ОПК-4, ПК-4	Тест	Низкий (неудовлетворительно)	Количество правильных ответов на вопросы теста менее 60 %
		Пороговый (удовлетворительно)	Количество правильных ответов на вопросы теста от 61-75 %
		Базовый (хорошо)	Количество правильных ответов на вопросы теста от 76-84 %
		Высокий (отлично)	Количество правильных ответов на вопросы теста от 85-100 %

6.2 Промежуточная аттестация студентов по дисциплине

Промежуточная аттестация является проверкой всех знаний, навыков и умений студентов, приобретённых в процессе изучения дисциплины. Формой промежуточной аттестации по дисциплине является экзамен.

Для оценивания результатов освоения дисциплины применяется следующие критерии оценивания.

Критерии оценивания устного ответа на экзамене

Экзаменационный билет содержит два теоретических вопроса. Кроме этого при выставлении оценки учитывается выполнение лабораторного практикума по дисциплине, задания которого размещены в Электронной информационно-образовательной среде БГПУ.

Оценка 5 (отлично) ставится, если:

- на оба вопроса даны исчерпывающие ответы, проиллюстрированные наглядными примерами там, где это необходимо;

- ответы изложены грамотным научным языком;
- все термины употреблены корректно, все понятия раскрыты верно;
- задания лабораторного практикума выполнены на 90 и более процентов;
- в ответе допущены 1-2 недочета.

Оценка 4 (хорошо) ставится, если:

- на вопросы даны в целом верные ответы, но с отдельными неточностями, не носящими принципиального характера;

- не все термины употреблены правильно;
- присутствуют отдельные некорректные утверждения и грамматические / стилистические погрешности изложения;

- ответы не проиллюстрированы примерами в должной мере.

- задания лабораторного практикума выполнены на 75 и более процентов.

Оценка 3 (удовлетворительно) ставится, если:

- ответы на вопросы носят фрагментарный характер, верные выводы перемежаются с неверными;
 - упущены содержательные блоки, необходимые для полного раскрытия темы;
 - студент в целом ориентируется в тематике учебного курса, но испытывает проблемы с раскрытием конкретных вопросов;
 - даны удовлетворительные ответы на оба вопроса экзаменационного билета и задания лабораторного практикума выполнены на 60 и более процентов;
 - дан удовлетворительный ответ только на один вопрос экзаменационного билета и задания лабораторного практикума выполнены на 75 и более процентов.
- Оценка 2 (неудовлетворительно) ставится, если:
- ответы на вопросы отсутствуют либо не соответствуют содержанию вопросов;
 - ключевые для учебного курса понятия, содержащиеся в вопросах, трактуются ошибочно;
 - задания лабораторного практикума выполнены менее чем на 60 процентов.

6.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов освоения дисциплины

Вопросы к собеседованию по теме «Виды диаграмм UML»

1. Разница между статическими и динамическими диаграммами.
2. Виды статических диаграмм.
3. Виды динамических диаграмм.
4. Способы изображения различных видов связей на диаграмме классов.
5. Отличия диаграмм последовательности и кооперации.
6. Сущности, отображаемые на диаграмме состояний.

Варианты разноуровневых задач и заданий Примеры задач для реализации программных проектов

1. Система бронирования билетов для авиакомпании

Краткое описание

На рынок вышла новая авиакомпания «GlobalAvia». Менеджеры компании решили заказать у вашей фирмы разработку системы бронирования билетов. При заказе фирма поставила ряд условий, которые обязательно должны быть выполнены. В первой версии системы они хотят видеть две части. Работа первой части системы связана с занесением информации. Вторая часть системы предназначена для общения с клиентами.

При формулировании требований менеджеры упомянули, что рейсы спланированы так, что до пункта назначения можно долететь с пересадками за разное время и с разным комфортом. Одно из требований заключалось в том, чтобы система помогала покупать билеты в зависимости от требований пользователя.

Полная постановка задачи

Задача является больше математической. Система должна уметь решать трехкритериальную задачу поиска кратчайших путей на графах. Критерии:

- Время.
- Цена.
- Комфорт.

Система является распределенной, поскольку в каждом аэропорту своя база направлений полетов самолетов, соответственно, знают о рейсе только аэропорты-соседи. Одно из требований, которое выдвигает компания, - не делать базу централизованной ввиду дороговизны оборудования, которое в противном случае пришлось бы приобрести.

Объекты системы: распределенное хранилище рейсов, покупатель билетов.

Распределенное хранилище рейсов: название рейсов, номера и тип самолетов, класс самолета по комфорту и стоимость билетов.

Покупатель: ФИО, сумма. Покупатель на сайте задает параметры, связанные с суммой, которую он хочет потратить, комфорт и время. Система должна подобрать оптимальные маршруты. При отсутствии прямых маршрутов система должна попробовать найти маршруты с пересадками. Если таковых не находится, система должна дать в ответе причину, по которой не получается подобрать маршрут. Среди причин:

- Отсутствие рейсов в требуемом направлении даже с пересадками.
- Сумма слишком мала.
- Комфорт превышен.

В ответ, пользователь должен иметь возможность поменять параметры с учетом предыстории.

Менеджер рейсов: должен иметь следующие возможности:

- создания и удаления аэропортов в системе.
- создания и удаления рейсов в аэропортах.

2. Редактор математических формул

Краткое описание

Фирма «OurResearch» занимается написанием математических программ по заказу. При этом в фирме часто приходится писать отчеты заказчику. При написании отчетов заказчик хочет видеть в отчетах математические формулы в классическом виде.

У Вашей фирмы компания решила заказать удобное средство для перевода и написания математических выражений в разные форматы представления. Причем, если в редакторе присутствует ряд взаимосвязанных формул, то фирма хочет видеть адекватный код. При этом известно, что фирма часто использует C/C++, Pascal и Fortran.

Полная постановка задачи

Необходимо разработать систему для редактирования и написания математических формул.

Объекты системы: формула, формульный редактор.

Формула: математическое выражение в одном из видов, желательно, чтобы редактор сам распознавал язык и вид выражения по сигнатуре.

Формульный редактор: визуальная часть. Должен позволять:

- транслировать стандартный синтаксис формул во внутренний формат;
- отображать из внутреннего формата в графический вид;
- визуально редактировать формулы;
- отображать структуру данных формулы.

Дополнительно система должна обеспечивать сохранение формул в нескольких форматах (например, в XML).

Редактор должен включать в себя конвертор в различные форматы, к примеру, перевод формулы в стандартное выражение для C/C++, Pascal и Fortran.

Также обязательно нужна возможность перевода формулы в BMP-изображение.

3. Web-сервис (на основе сокетов)

Краткое описание

Необходимо реализовать на стороне сервера хранилище, в которое можно помещать алгоритмы в некотором стандартном виде, а потом исполнять их. Для простоты алгоритмы могут представлять собой математические формулы. В алгоритмах должны быть заявлены следующие данные:

- входные данные;
- выходные данные;
- код алгоритмов.

Доступ к алгоритмам должен быть ограничен на основе разделения прав по ролям.

Полная постановка задачи

Web-сервис должен быть рассчитан на небольшое число пользователей и работу в локальной сети. К web-сервису должен быть реализован разделенный доступ пользователей.

Объекты системы: пользователь, роль, алгоритм, web-сервис.

Пользователи: логин, пароль, роль.

Пользователь может на web-сервисе осуществлять следующие действия:

- размещать алгоритмы;
- изымать на редактирование алгоритмы;
- удалять алгоритмы с web-сервиса;
- исполнять алгоритмы.

Роль: представляет собой список пользователей.

Алгоритм: название, код (математическое выражение), принадлежность пользователю, входные и выходные параметры.

Web-сервис предоставляет следующие возможности:

- хранить алгоритмы на сервере;
- предоставлять доступ к алгоритмам:
 - редактирование;
 - удаление;
- исполнение алгоритма на сервере.

Для хранения алгоритмов на сервере создается дерево каталогов и файлов. Для каждого пользователя создается корневой каталог. В этом каталоге могут храниться, как алгоритмы (файлы с кодом), так и другие каталоги. Разделение прав осуществляется на основе специального файла со списком пользователей, которым доступна эта папка. Права на папку наследуются. Также можно разрешить доступ сразу группе пользователей, принадлежащих определенной роли.

Редактировать алгоритм может только пользователь, выложивший алгоритм.

Исполнить алгоритм могут только те пользователи, которым доступна папка с алгоритмом. Исполнение производится через специальный интерфейс.

4. Система взаимодействия команд

Краткое описание

Руководство фирмы “Effectiveness” пришло к выводу, что производительность труда ее сотрудников недостаточно полно соответствует громкому имени компании. В результате проведенных исследований была выявлена основная причина – недостаточная эффективность обмена информацией между сотрудниками. В качестве решения руководство видит внедрение единого средства электронной коммуникации. Отделу разработчиков компании поручено создать систему, включающую в себя почтовый инструмент и инструмент для обмена мгновенными сообщениями.

Полная постановка задачи

Требуется реализовать комплексную систему обмена сообщениями для небольшого количества пользователей.

Система должна удовлетворять следующим требованиям:

- возможность отправки и получения почты;
- возможность чата с пользователями, находящимися в сети.

Объекты системы: пользователь, сервер, письмо, список контактов, быстрое сообщение, почтовый клиент, клиент для чата.

Пользователь: ФИО, логин, пароль.

Сервер: представляет собой хранилище почты и маршрутизатор сообщений. Сервер должен разграничивать доступ к информации различных пользователей. Каждый пользователь может получить только свою почту. Пользователь может присоединиться к чатам и участвовать в обсуждении, но только в том случае, если он входит в список тех, кому разрешен доступ.

Письмо: для написания писем предполагается разработка простейшего текстового редактора. Внутренний формат – некоторая динамическая структура данных. При сохранении и отправке должен генерироваться HTML-документ. При приеме HTML-документ должен разбираться во внутреннюю структуру. Предполагается ограниченный набор тегов и свойств тегов.

Быстрое сообщение: обычное текстовое сообщение, для генерации которого используется редактор письма. Отличие от почты в том, что сообщение видят все, находящиеся в чате, сразу после отправки.

Список контактов: список логинов пользователей системы.

Почтовый клиент: включает в себя редактор почты, средства просмотра и отправки почты лицам из списка контактов.

Клиент для чата: содержит список доступных чатов. Предоставляет возможность создания чатов и открытие чата для людей из списка контактов.

5. Учет работы персонала

Краткое описание

В компании “Justice” были проведены исследования, в результате которых компания поняла, что теряет достаточно много средств из-за не всегда равномерного распределения зарплаты. Кроме денежных потерь есть и недовольство персонала, которое вызвано тем, что те, кто много работают, и те, кто мало, получают одинаковые зарплаты.

Компания решила изменить данную ситуацию. Вашей компании предложили выработать решение описанной проблемы.

Полная постановка задачи

Нужно реализовать систему учета работы персонала.

Объекты системы: датчики, хранилища данных, менеджер, работник, система начисления зарплаты, система построения отчетов.

Датчики: считается, что фирма состоит из нескольких филиалов и в каждом филиале есть отдельные наборы датчиков. Каждый набор датчиков сохраняет информацию в свое хранилище данных. Основные данные, с которыми работает система:

- Кто и во сколько пришел.
- Кто и во сколько ушел.
- Кто, когда отбыл в командировку.
- Кто и когда вернулся из командировки.

Хранилище данных: является распределенным. Предназначение – вести журнал данных, поступающих с датчиков. Доступ к хранилищу могут получать только **работники и менеджеры**.

В системе должны существовать две роли: **работник** и **менеджер**. При этом менеджер также может быть и работником.

Работник: ФИО, логин, пароль, табельный номер, счет для начисления зарплаты. При каждом посещении филиала записывается время прибытия отбытия и, возможно, работа, которую он выполнял. Работники могут просмотреть, сколько и когда они работали, и какая зарплата их ожидает.

Менеджер: контролирующая должность. Содержит ФИО, логин, пароль, табельный номер, счет для начисления зарплаты.

Выполняет следующие функции:

- Построение и просмотр отчетности по работникам (через **систему построения отчетов**).

- Премирование выделившихся работников.
- Прием и увольнение персонала.
- Поднятие и понижение разрядов и определение квалификационной группы.
- Установление индексов зарплат для работников.
- Контроль за системой начисления зарплаты.

Система начисления зарплаты: для автоматизации необходимо два раза в месяц делать начисление зарплаты работнику. При расчете зарплаты в системе часы, проведенные работником на месте, умножаются на индекс. К полученной сумме прибавляются премиальные и из получившейся суммы вычитаются налоги. Итоговая сумма переводится на счет.

6. Система управления проектами

Краткая постановка

В компании “SuperSoft” возникла потребность автоматизировать управление проектами. В силу того, что компания существует на рынке разработки ПО недавно и не обладает достаточным количеством свободных финансовых средств, было принято решение не покупать системы управления проектами типа Microsoft Project (стоимость коробочной версии от \$600), а разработать собственное простое решение.

Система управления проектами должна иметь единую базу проектов, подключаться к которой могут менеджеры и исполнители. Содержимое базы составляет информация о ведущихся в компании проектах.

Полная постановка задачи

Система управления проектами должна быть рассчитана на небольшие команды. В каждом проекте выделяются только две роли: менеджер и исполнитель.

Менеджер может управлять несколькими проектами, исполнитель участвует только в одном проекте.

Менеджер управляет проектом, то есть с точки зрения системы: формирует список задач проекта, распределяет задачи по исполнителям (ограничение: нет задач, предназначенных более чем одному исполнителю), формирует план-график выполнения проекта (задает сроки выполнения задач), выставляет состояние задач (не начата, выполняется, завершена, отложена).

Исполнитель получает от менеджера задачи, на основании которых для него формируется “ToDo-List”. Список может пополняться по ходу проекта, как менеджером, так и самим исполнителем.

Объекты системы: менеджер, исполнитель, задача, проект, “ToDo-List”.

Менеджер: ФИО, проект(ы).

Исполнитель: ФИО, проект, “ToDo-List”.

Задача: имя, формулировка, срок начала, срок окончания (выставленный менеджером), срок фактического окончания (когда исполнитель “сдал” задачу менеджеру, а тот ее “принял”), состояние (не начата, выполняется, завершена, отложена), причина изменения срока окончания/откладывания.

Проект: имя, менеджер, исполнители, задачи.

“ToDo-List”: список задач для исполнителя. Часть списка формируется автоматически, доступна только для чтения. Часть формируется исполнителем “для себя”, доступна для редактирования.

Сроки для задач могут задаваться с точностью до часов (начало: 21 июля 14.00, окончание: 21 июля 16.00, фактического окончания: 21 июля 17 часов, причина изменения сроков: учения по пожарной безопасности).

Хранение всех данных централизовано. Система имеет серверную часть (хранение информации и интерфейс для администрирования) и клиентскую часть, с которой работают менеджеры и исполнители.

По данным каждого проекта в системе должна быть возможность поиска.

Темы для докладов в лабораторной работе № 1 Гибкие методологии разработки программного обеспечения

1. XP
2. Crystal
3. Scrum
4. Kanban
5. ASD
6. FDD
7. TDD и BDD
8. MDD
9. DSDM
10. Lean

Вопросы к экзамену

1. Термины программное обеспечение, программирование, профессиональное программирование, программная инженерия.
2. Эволюция системного программного продукта и неотъемлемые свойства программного обеспечения по Ф. Бруксу.
3. Компоненты системы разработки программного обеспечения, выделенные Э. Брауде.
4. Определение технологии разработки программного обеспечения. Этапы разработки программного обеспечения. CASE-система: определение, примеры.
5. Этапы эволюции технологии разработки программ. OOA, OOD, OOP.
6. Жизненный цикл программного обеспечения. Процессы жизненного цикла.
7. Модели жизненного цикла программного обеспечения.
8. Требования к программному обеспечению. Определение. Виды.
9. Этапы процесса управления требованиями.
10. Термины архитектура программного обеспечения, компонент программного обеспечения.
11. Разработка архитектуры программного обеспечения на основе сценариев использования.
12. Диаграммы UML.
13. Виды программной документации. Стандарты, устанавливающие правила разработки программной документации.
14. Качество программного обеспечения. Определение, виды.
15. Характеристики качества программного обеспечения и их атрибуты.
16. Метрики оценки атрибутов качества программного обеспечения. Виды, примеры.
17. Верификация программного обеспечения. Определение, цели и задачи.
18. Тестирование программного обеспечения. Определение, типы. Тестовые данные, тестовая ситуация, удачный тест, ошибка.
19. Тестирование программного кода. Методы тестирования. Тестовое окружение, драйверы и заглушки, подходы к тестированию классов.
20. Тестовые примеры, тест-требования, тест-план. Типы тестовых примеров. Классы эквивалентности.
21. Сбой, отказ, авария. Виды сбоев, виды отказов. Методы разработки устойчивого кода.
22. Основные этапы проектирования пользовательского интерфейса.
23. Факторы оценки удобства использования программного обеспечения.
24. Правила и принципы создания удобного пользовательского интерфейса.
25. Задача управления проектом. Основные виды ресурсов проекта.
26. Виды деятельности, входящие в управление проектом.
27. Коллективная разработка программного обеспечения. Организационный аспект, инструментальные средства поддержки.

7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ В ПРОЦЕССЕ ОБУЧЕНИЯ

Информационные технологии – обучение в электронной образовательной среде с целью расширения доступа к образовательным ресурсам, увеличения контактного взаимодействия с преподавателем, построения индивидуальных траекторий подготовки, объективного контроля и мониторинга знаний студентов.

В образовательном процессе по дисциплине используются следующие информационные технологии, являющиеся компонентами Электронной информационно-образовательной среды БГПУ:

- официальный сайт БГПУ;
- корпоративная сеть БГПУ;
- система электронного обучения ФГБОУ ВО «БГПУ»;
- электронные библиотечные системы;
- мультимедийное сопровождение лекций и практических занятий;
- программа для создания диаграмм UML;
- интегрированная среда разработки программного обеспечения.

8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

При обучении лиц с ограниченными возможностями здоровья применяются адаптивные образовательные технологии в соответствии с условиями, изложенными в раздел «Особенности организации образовательного процесса по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья» основной образовательной программы (использование специальных учебных пособий и дидактических материалов, специальных технических средств обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь и т.п.) с учётом индивидуальных особенностей обучающихся.

9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ

9.1 Литература

1. Галиаскаров Э. Г. Анализ и проектирование систем с использованием UML : учебное пособие для вузов / Э. Г. Галиаскаров, А. С. Воробьев. – Москва: Издательство Юрайт, 2022. – 125 с. – (Высшее образование). – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/viewer/analiz-i-proektirovanie-sistem-s-ispolzovaniem-uml-497207> (дата обращения: 06.09.2022).

2. Гниденко И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Издательство Юрайт, 2022. – 235 с. – (Высшее образование). – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/viewer/tehnologii-i-metody-programmirovaniya-489920> (дата обращения: 06.09.2022).

3. Зуб А. Т. Управление проектами: учебник и практикум для вузов / А. Т. Зуб. – Москва: Издательство Юрайт, 2022. – 422 с. – (Высшее образование). – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/viewer/upravlenie-proektami-489197> (дата обращения: 06.09.2022).

4. Иванова, Г.С. Технология программирования : учеб / Г.С. Иванова. – М.: Кнорус, 2011. – 333 с. (6 экз.)

5. Лаврищева Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2022. – 432 с. – (Высшее образование). – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/viewer/programmnyaya-inzheneriya-i-tehnologii-programmirovaniya-slozhnyh-sistem-491029> (дата обращения: 06.09.2022).

6. Лаврищева Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства : учебник для вузов / Е. М. Лаврищева. – 2-е изд., испр. – Москва: Издательство Юрайт, 2022. – 280 с. – (Высшее образование). – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/viewer/programmnyaya-inzheneriya-paradigmy-tehnologii-i-case-sredstva-491048> (дата обращения: 06.09.2022).

7. Чекмарев А. В. Управление ИТ-проектами и процессами : учебник для вузов / А. В. Чекмарев. – Москва: Издательство Юрайт, 2022. – 228 с. – (Высшее образование). – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/viewer/upravlenie-it-proektami-i-processami-493916> (дата обращения: 06.09.2022).

8. Черткова Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2022. – 147 с. – (Высшее образование). – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/viewer/programmnyaya-inzheneriya-vizualnoe-modelirovanie-programmnyh-sistem-491629> (дата обращения: 06.09.2022).

9.2 Базы данных и информационно-справочные системы

1. Федеральный портал «Открытое образование». – Режим доступа: <https://openedu.ru>

2. Электронный фонд нормативно-технической и нормативно-правовой документации Консорциума «Кодекс». – Режим доступа: <https://docs.cntd.ru/>

9.3 Электронно-библиотечные ресурсы

1. ЭБС «Юрайт». – Режим доступа: <https://urait.ru>

2. Полпред (обзор СМИ). – Режим доступа: <https://polpred.com/news>

10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Для проведения лекционных и лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используются аудитории, оснащённые учебной мебелью, аудиторной доской, компьютером с установленным лицензионным специализированным программным обеспечением, с выходом в электронно-библиотечную систему и электронную информационно-образовательную среду БГПУ, мультимедийными проекторами, экспозиционными экранами, учебно-наглядными пособиями (мультимедийные презентации).

Для проведения лабораторных занятий также используются компьютерные классы, укомплектованные следующим оборудованием:

- Комплект столов письменных.
- Стол преподавателя.
- Аудиторная доска.

- Компьютеры с установленным лицензионным специализированным программным обеспечением.
- Мультимедийный проектор.
- Экспозиционный экран.
- Учебно-наглядные пособия – мультимедийные презентации по дисциплине «Технология разработки программного обеспечения».

Самостоятельная работа студентов организуется в аудиториях оснащенных компьютерной техникой с выходом в электронную информационно-образовательную среду вуза, в специализированных лабораториях по дисциплине, а также в залах доступа в локальную сеть БГПУ.

Лицензионное программное обеспечение: операционные системы семейства Windows, Linux; офисные программы Microsoft office, Libreoffice, OpenOffice; Adobe Photoshop, Matlab, DrWeb antivirus и т.д.

Разработчик: Апалеева А.М. – преподаватель кафедры информатики и методики преподавания информатики

11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ

Утверждение изменений и дополнений в РПД для реализации в 2020/2021 уч. г.

РПД обсуждена и одобрена для реализации в 2020/2021 уч. г. на заседании кафедры информатики и методики преподавания информатики (протокол № 8 от «17» июня 2020 г.). В РПД внесены следующие изменения и дополнения:

№ изменения: 1 № страницы с изменением: Титульный лист	
Исключить:	Включить:
Текст: Министерство науки и высшего образования РФ	Текст: Министерство просвещения Российской Федерации

Утверждение изменений и дополнений в РПД для реализации в 2021 /2022 уч. г.

РПД обсуждена и одобрена для реализации в 2021/2022 уч. г. на заседании кафедры информатики и методики преподавания информатики (протокол № 1 от «8» сентября 2021 г.). В РПД внесены следующие изменения и дополнения:

№ изменения: 2 № страницы с изменением: 3	
В п 1.3 ОПК-3. изложить в следующей редакции: ОПК-3. Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения.	

Утверждение изменений и дополнений в РПД для реализации в 2022/2023 уч. г.

РПД пересмотрена, обсуждена и одобрена для реализации в 2022/2023 учебном году на заседании кафедры информатики и методики преподавания информатики (протокол №1 от 21 сентября 2022 г.).

В рабочую программу внесены следующие изменения и дополнения:

№ изменения: 1 № страницы с изменением: 21-22	
В Раздел 9 внесены изменения в список литературы, в базы данных и информационно-справочные системы, в электронно-библиотечные ресурсы. Указаны ссылки, обеспечивающие доступ обучающимся к электронным учебным изданиям и электронным образовательным ресурсам с сайта ФГБОУ ВО «БГПУ».	

Утверждение изменений и дополнений в РПД для реализации в 2024/2025 уч. г.

РПД обсуждена и одобрена для реализации в 2024/2025 уч. г. без изменений на заседании кафедры информатики и методики преподавания информатики (протокол №8 от 30.05.2024 г.).